CLAIMS:

The following is a listing of all claims in the application with their status and the text of all active claims:

1.-9. (CANCELED)

10. (NEW) An apparatus for routing address lookup in a router, comprising of at least one monolithic integrated device, each comprising:

(a) a plurality of content addressable memory banks for storing and looking up the routing addresses,

(b) at least one bank-level content comparing memory array for each content addressable memory bank, said bank-level content comparing memory array or arrays storing the range of the routing addresses in the corresponding content addressable memory bank,

(c) first means for interpreting the contents of said bank-level content comparing memory array or arrays for deciding whether the search address is in the corresponding content addressable memory bank,

(d) second means for connecting said bank-level content comparing memory array or arrays to said content addressable memory bank and for activating said content addressable memory bank based on the interpretation of said first means,

whereby, any particular memory bank of said content addressable memory banks is only activated if the routing address is deterministically known to be in that memory bank, which saves the power consumed by the apparatus.

11. (NEW) The apparatus of claim 10, wherein each content addressable memory bank is associated with a routing address prefix length and each routing address prefix length is associated with at least one of said content addressable memory banks,

whereby,

- said content addressable memory banks can be of binary type rather than ternary type, which makes it smaller in size,
- the need to have a priority encoder inside each content addressable memory bank to resolve multiple prefix length matches is eliminated, resulting in a higher operation speed.

12. (NEW) The apparatus of claim 11, further including a mask memory array, which specifies the routing address prefix length that each content addressable memory bank is associated to.

13. (NEW) The apparatus of claim 10, further including:

(a) a plurality of chip-level content comparing memory arrays corresponding to each monolithic integrated device, said chip-level content comparing memory arrays storing the range of the routing addresses in the corresponding monolithic integrated device,

(b) first means for interpreting the contents of said chip-level content comparing memory arrays for deciding whether the search address is in the corresponding monolithic integrated device,

(c) second means for connecting said chip-level content comparing memory arrays to said bank-level content comparing memory arrays and for activating said bank-level content comparing memory arrays based on the interpretation of said first means,

whereby, any particular monolithic integrated device is only activated if the routing address is deterministically known to be in that device, which saves the power consumed by the apparatus.

14. (NEW) The apparatus of claim 13, wherein each content addressable memory bank is associated with a routing address prefix length and each routing address prefix length is associated with at least one of said content addressable memory banks,

    whereby,

    - said content addressable memory bank can be of binary type rather than ternary type, which makes it smaller in size,
    - the need to have a priority encoder inside each content addressable memory bank to resolve multiple prefix length matches is eliminated, resulting in higher operation speed.

15. (NEW) The apparatus of claim 14, further including a mask memory array, which specifies the routing address prefix length that each content addressable memory bank is associated to.

16. (NEW) A content comparing memory device for generating the carry bit or bits in the summation of a search binary word to at least one stored binary word, comprising a plurality of content comparing memory cells arranged in rows and columns, with each stored binary word stored in each of said rows, each of said content comparing memory cells comprising:

    (a) a normal memory cell for storing the stored binary bit,

    (b) means for reading from and writing to said normal memory cell,

    (c) a signal line for delivering the search binary bit,

(d) a first logic device which provides a carry transfer logical operation selected from the group consisting of logical XOR and logical OR of said search binary bit or its inverse and said stored binary bit or its inverse,

(e) a second logic device which provides logical AND of said search binary bit or its inverse and said stored binary bit or its inverse,

(f) an input port for delivering the carry-in value for the bit summation,

(g) an output port for delivering the carry-out value of the bit summation,

(h) said input port connecting to the output port of the previous content comparing memory cell in the row and said output port connecting to the input port of the next content comparing memory cell in the row,

(i) said first logic device driving a passgate between said input port and said output port, said passgate transferring the carry-in value to said output port when turned on,

(j) said second logic device driving said output port to a predetermined carry logical value indicating carry bit in the summation of said stored binary bit or its inverse and said stored binary bit or its inverse,

whereby, the output port of the last content comparing memory cell in a row is driven to the said carry logical value if the summation of the stored binary word of said row, said search binary word, and the carry-in value applied to the input port of the first content comparing memory cell in said row generates a carry.

17. (NEW) Content comparing memory device of claim 16, wherein said normal memory cell stores both said stored bit and its logical inverse.

18. (NEW) Content comparing memory device of claim 16, further including an inverter for creating the logical inverse of said stored bit in each of said content comparing memory cells.

19. (NEW) Content comparing memory device of claim 16, further including a signal line for delivering the inverse of said search binary bit in each of said content comparing memory cells.

20. (NEW) Content comparing memory device of claim 19, wherein said normal memory cell stores both said stored bit and its logical inverse.

21. (NEW) Content comparing memory device of claim 19, further including an inverter for creating the logical inverse of said stored bit in each of said content comparing memory cells.

22. (NEW) Content comparing memory device of claim 16 wherein said first logic device and said second logic device are made from transmission gates, whereby, the device area becomes smaller.

23. (NEW) Content comparing memory device of claim 22, wherein said normal memory cell stores both said stored bit and its logical inverse.

24. (NEW) Content comparing memory device of claim 22, further including an inverter for creating the logical inverse of said stored bit in each of said content comparing memory cells.

25. (NEW) Content comparing memory device of claim 22, further including a signal line for delivering the inverse of said search binary bit in each of said content comparing memory cells.

26. (NEW) Content comparing memory device of claim 25, wherein said normal memory cell stores both said stored bit and its logical inverse.

27. (NEW) Content comparing memory device of claim 25, further including an inverter for creating the logical inverse of said stored bit in each of said content comparing memory cells.

28. (NEW) A method for routing address lookup in a router, comprising:

(a) providing at least one monolithic integrated device which contains means for storing the routing addresses and performing longest prefix match in memory banks,

(b) considering the collection of all stored routing addresses as a range of addresses, in this range representing an m bit routing address with a prefix length of n with its full prefix length representation, which is a range of 2^{m-n} consecutive addresses,

(c) dividing said range of addresses to a number of non-overlapping bank-level address ranges, assigning each bank-level address range to one of said memory banks, storing in each memory bank only the routing addresses that fall within its assigned bank-level address range,

(d) storing routing addresses for which the full prefix length representation span multiple bank-level address ranges, on all the corresponding memory banks,

(e) providing means for comparing the search routing address with said bank-level address ranges, and activating the memory banks that their bank-level ranges contain said search routing address, if any,

whereby, any particular memory bank of said monolithic integrated device or devices will be only activated if said search routing address is deterministically known to be in that memory bank, which saves the power consumed for the routing address lookup.

29. (NEW) The method of claim 28, further including:

(a) providing means for comparing said search routing address with the chip-level address ranges, each chip-level address range corresponding to one monolithic integrated device and being the collective address range of all said memory banks on said monolithic integrated device,

(b) activating the monolithic integrated devices that their chip-level address ranges contain said search routing address, if any,

whereby, any particular monolithic integrated device will be only activated if said search routing address is deterministically known to be in said monolithic integrated device, which saves the power consumed for the routing address lookup.

30. (NEW) A method for routing address lookup in a router, comprising:

(a) providing at least one monolithic integrated device which contains means for storing the routing addresses and performing longest prefix match in memory banks,

(b) considering the collection of all stored routing addresses as a range of addresses, in this range representing an m bit routing address with a prefix length of n with its full prefix length representation, which is a range of 2^{m-n} addresses,

(c) dividing said range of addresses to a number of non-overlapping chip-level address ranges, assigning each chip-level address range to one of said monolithic integrated devices, if there are more than one, storing on each monolithic integrated device only the routing addresses that fall within its assigned chip-level address range,

(d) storing routing addresses for which the full prefix length representation span multiple chip-level address ranges, if there are more than one, on all the corresponding monolithic integrated devices,

(e) dividing said memory banks to a number of prefix length bank groups equal to the number of distinctive prefix lengths in said routing addresses, assigning each distinctive prefix length to one of said prefix length bank groups, storing in each prefix length bank group only routing addresses with its assigned prefix length,

(f) for each prefix length bank group, dividing each chip-level address range to a number of bank-level address ranges, assigning each bank-level address range to one of said memory banks, storing in each bank only the routing addresses that fall within its assigned bank-level range and have the prefix length assigned to the prefix length bank group,

(g) providing means for comparing the search routing address with said chip-level address ranges and activating the monolithic integrated device that its chip-level address range contains said search routing address, if any,

(h) providing means for comparing said search routing address with the bank-level address ranges inside the activated monolithic integrated device, if any, and activating the memory banks that their bank-level address ranges contain said search routing address, if any,

whereby,

- any particular monolithic integrated device and any particular memory bank of said monolithic integrated device will be only activated if said search routing address is

deterministically known to be in that memory bank of that monolithic integrated device, which saves the power consumed for the routing address lookup,

- the need to have a priority encoder inside each memory bank to resolve multiple prefix length matches is eliminated, resulting in higher operation speed.

31. (NEW) A method for routing address lookup in a router, comprising:

(a) providing a device which contains means for storing the routing addresses and performing longest prefix match, said device consisting of at least one hierarchy level of routing address storage and search units,

(b) considering the collection of all stored routing addresses as a range of addresses, in this range representing an m bit routing address with a prefix length of n with its full prefix length representation, which is a range of 2^{m-n} addresses,

for each hierarchy level:

(c) dividing said range of addresses to a number of non-overlapping unit-level address ranges, assigning each unit-level address range to one of said storage and search units in the hierarchy level, storing in said storage and search unit only the routing addresses that fall within its assigned unit-level range,

(d) storing routing addresses for which said full prefix length representation span multiple unit-level address ranges on all the corresponding storage and search units,

(e) providing means for comparing the search routing address with said unit-level address ranges, and activating the storage and search units that their unit-level address range contains said search routing address, if any,

whereby, any particular storage and search unit in each hierarchy level will be only activated if said search routing address is deterministically known to be in that storage and search unit, which saves the power consumed for the routing address lookup.

32. (NEW) A method for comparing a search binary word to a stored binary word, comprising:

(a) providing a content comparing memory array of same length as said stored binary word which generates, at its output, the carry-out bit for the summation of a binary word applied to it, the binary word stored in the array, and a carry-in value applied at its input,

(b) setting a logical carry-in value of 0 at said input of said content comparing memory,

(c) selecting from the group consisting of:

storing the 2's complement of said stored binary word in said content comparing memory array,

and,

applying the 2's complement of said search binary word to said content comparing memory array,

(d) observing the fact that when a number x is added to the 2's complement of a number y, said carry-out bit will be 1 if x>y and it will be 0 if x<y,

whereby, from said output of said content comparing memory it can be judged whether said search binary word is larger than said stored binary word or not.

33. (NEW) A method for comparing a search binary word to a stored binary word, comprising:

    (a) providing a content comparing memory array of same length as said stored binary word which generates, at its output, the carry-out bit for the summation of a binary word applied to it, the binary word stored in the array, and a carry-in value applied at its input,

    (b) setting a logical carry-in value of 1 at said input of said content comparing memory,

    (c) selecting from the group consisting of:

        storing the bitwise inversion of said stored binary word in said content comparing memory array,

        and,

        applying the bitwise inversion of said search binary word to said content comparing memory array,

    (d) observing the fact that when binary number 1 is added to the summation of a number x and the bitwise inversion of a number y, said carry-out bit will be 1 if x>y and it will be 0 if x<y,

whereby, from said output of said content comparing memory it can be judged whether said search binary word is larger than said stored binary word or not.

34. (NEW) A method for comparing a search binary word to a stored binary word, comprising:

    (a) providing a content comparing memory array of same length as said stored binary word which generates an output selected from the group consisting of:

the carry-out bit for the summation of a binary word applied to it, the bitwise inversion of the binary word stored in the array, and a carry-in value applied at its input,

and,

the carry-out bit for the summation of the bitwise inversion of a binary word applied to it, the binary word stored in the array, and a carry-in value applied at its input,

(b) setting a logical carry-in value of 1 at said input of said content comparing memory,

(c) storing said stored binary word as it is in said content comparing memory array,

(d) observing the fact that when binary number 1 is added to the summation of a number x and the bitwise inversion of a number y, said carry-out bit will be 1 if $x>y$ and it will be 0 if $x<y$,

whereby, from said output of said content comparing memory it can be judged whether said search binary word is larger than said stored binary word or not.